Introducción a la Bioinformática:

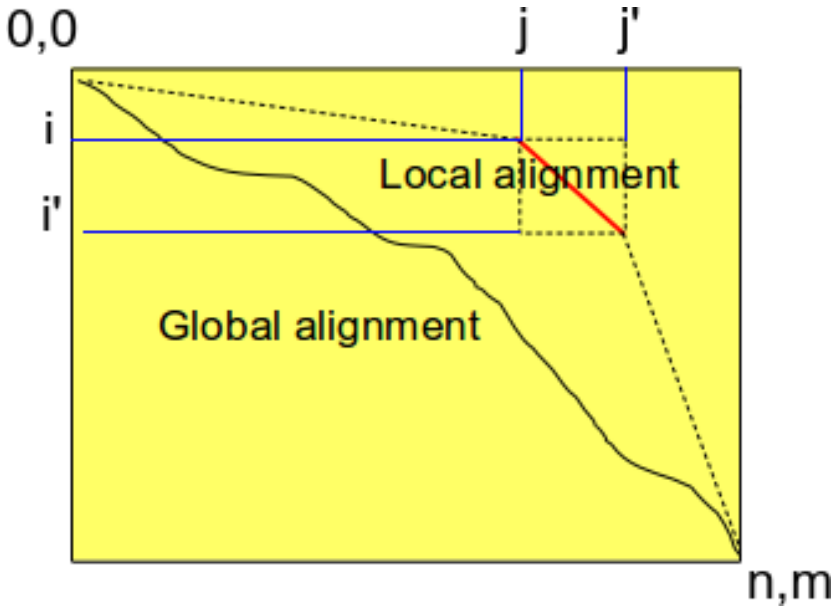
Comparative Genomics: Sequence Alignments I

Luis Garreta

Doctorado en Ingeniería
Pontificia Universidad Javeriana – Cali

April 7, 2018

Local vs. Global Alignment



- The Global Alignment Problem tries to find the optimal alignment between position (0,0) and (n,m) in the Dynamic Programming matrix.
- The Local Alignment Problem tries to find the optimal alignment among alignments between arbitrary positions (i,j) and (i', j') in the Dynamic Programming matrix.

Local vs. Global Alignment

- Global Alignment

```
--T--CC-C-AGT--TATGT-CAGGGGACACG-A-GCATGCAGA-GAC
  |   || |   ||   | | | |||     || | | |   | ||||    |
AATTGCCGCC-GTCGT-T-TTCAG----CA-GTTATG-T-CAGAT--C
```

- Local Alignment

```
                    tccCAGTTATGTCAGggggacacgagcatgcagagac
                       ||||||||||||
aattgccgccgtcgttttcagCAGTTATGTCAGatc
```

Local Alignments are better to find conserved segment:

- Identify regions of high similarity between two sequences

Dynamic Programming Matrix:
For each step

Scoring Function

```
Match:      +1 (s)
Mismach:     0 (s)
Indel:      -1 (d)
```

Compute the max score for each cell:

- According to the scoring function
- According to the neighbors

F:

Pos		1	2	3	4	5	6
		–	**C**	**A**	**C**	**G**	**A**
1	–	0	-1	-2	-3	-4	-5
2	**C**	-1	↘ +1 ↓ -1 → -1 **1**				
3	**G**	-2					
4	**A**	-3					

Needleman-Wunch Algorithm (Global Alignment)

Conventions

F: DP Matrix **NxM**

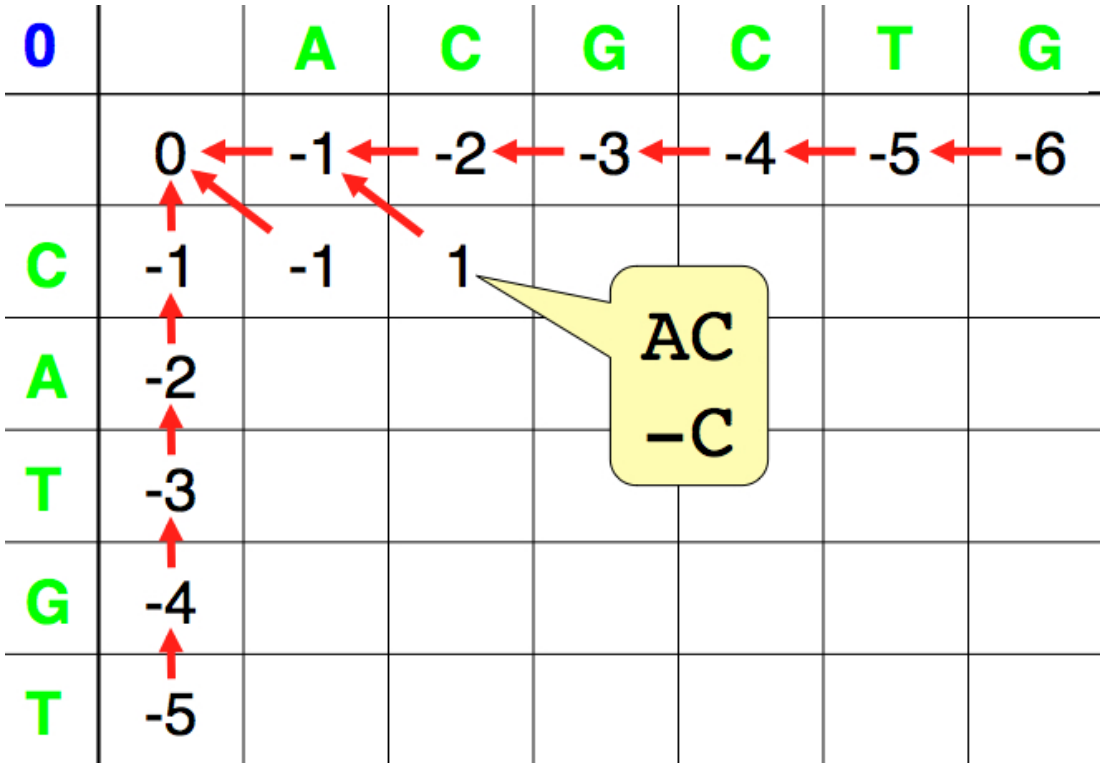
s(x,y): for match and mismatch at x,y

d: for indel events (gap penalty)

1. Initializacion (two sequences of length M and N)
   - 1.1 F (0, 0) = 0
   - 1.2 F (0, j) = -j*d
   - 1.3 F (i, 0) = -i*d
2. Main Iteration (Filling-in partial alignments)
   - 2.1 For each i=1 ..... M
   - 2.2 For each j=1 ..... N
   - 2.3 F(i,j) = max (F(i-1, j) - d, F(i, j-1) - d, F(i-1, j-1) + s(x,y))
3. Termination F(M,N) is the optimal score
4. Traceback for the optimal alignment

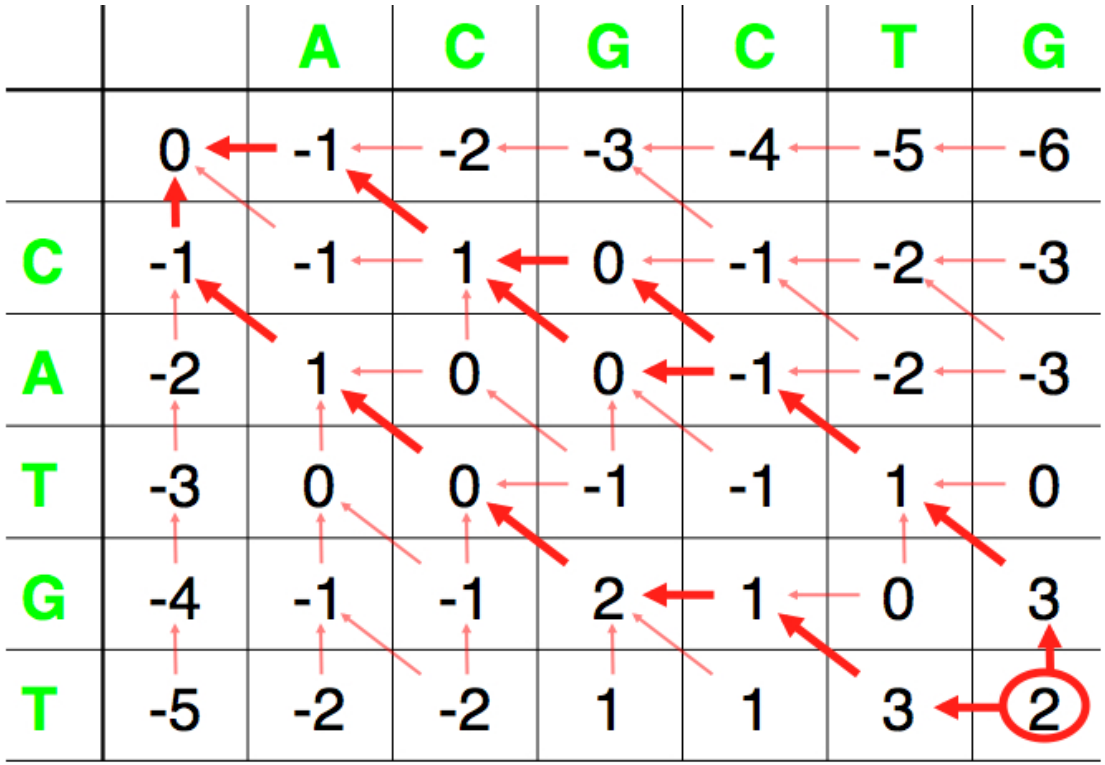
Example: Global Alignment for ACGCTG and CATGT

- Gap Penalty = -1, match = +2, mismatch = -1



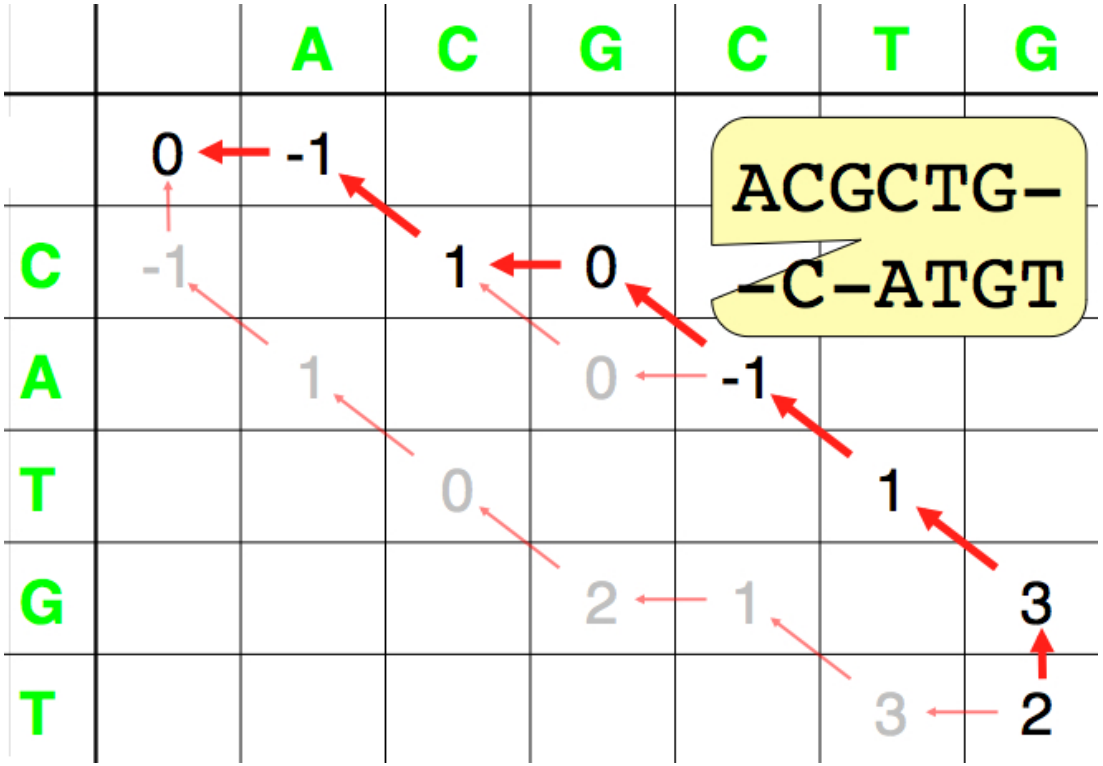
Example: Global Alignment for ACGCTG and CATGT

- Gap Penalty = -1, match = +2, mismatch = -1



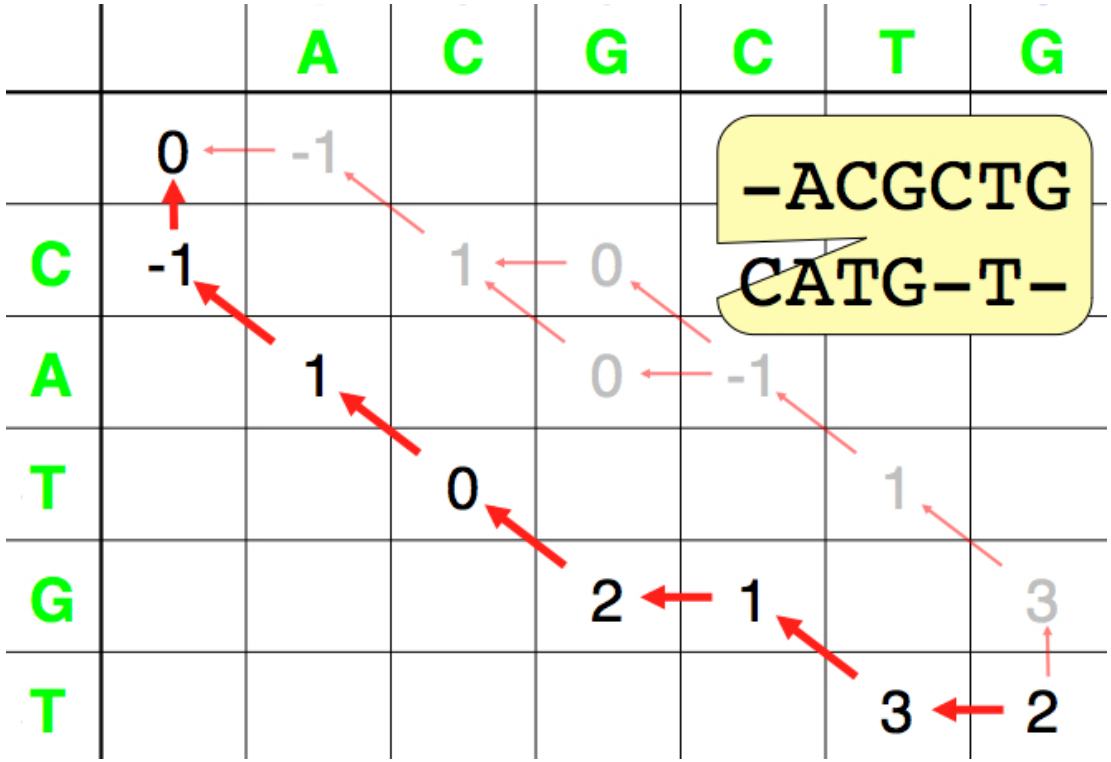
Example: Global Alignment for ACGCTG and CATGT

- Gap Penalty = -1, match = +2, mismatch = -1



Example: Global Alignment for ACGCTG and CATGT

- Gap Penalty = -1, match = +2, mismatch = -1



Considerations of Global Alignments

- Global alignments compares two sequences in their entirety
- Gap penalty is assessed regardless of whether gaps are located:
  - within a sequence
  - or at the end of one or both sequences
- This is not always the most desirable way to align two sequences

Considerations of Global Aligments

- For example: AACACGTGTCT and ACGT
  - match = 1, mismatch=0, gap -1

```
AACACGTGTCT
-AC--GT----

Global Score = 4 - 7 = -3
```

```
AACACGTGTCT
---ACGT----

Global score = 4 - 7 =  -3
```

- What has more biological meaningful?

No penalty for end gaps
AACACGTGTCT and ACGT

- However, of the several possible aligments, we are most interesting in is:

```
AACACGTGTCT
---ACGT----
```

- It demostrates that the shorter sequence appears in its entirety within the longer sequence
- To get this kind of alignments, we need to avoid penalizing for gaps that appear at one or both ends of a sequence
  - To treat them differently than internal gaps

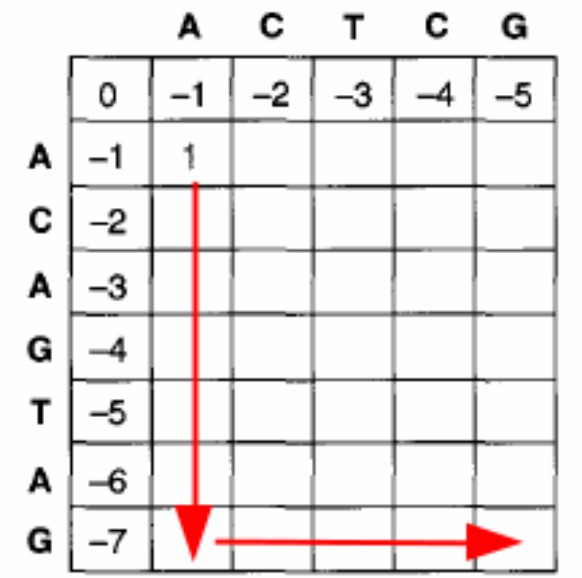
Terminal Gaps are usually the result of an incomplete data acquisition and do not have biological significance

Semiglobal Aligments

Modified Needleman and Wunsh Algorithm

Our original algorithm:

- Initialize the first column and row of the table with multiples of the gap penalty



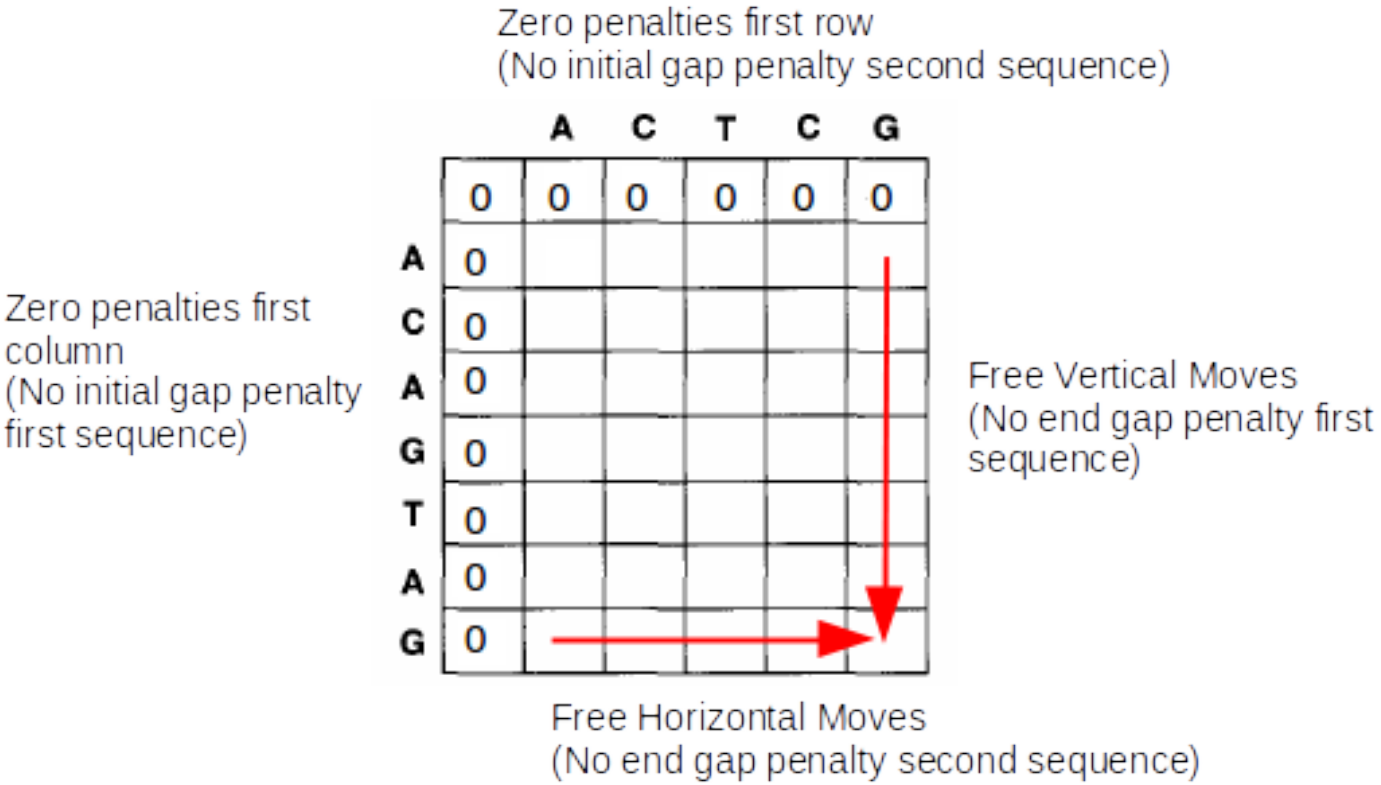
- By moving vertically to the bottom of the table and then horizontally to the rightmost edge:

```
-------ACTCG
ACAGTAG-----
```

Semiglobal Alignments

Initial and end gaps with no penalty:

- To allow initial gaps in the first or second sequence, we initialize the first column and row with no penalty (to all zeros)
- To allow end gaps to the first or second sequence, we allow free horizontal and vertical moves in the last row and column



Example: Semiglobal Aligment
ACACTGATCG and ACACTG

- Match=1, Mismatch=0, Gap=-1

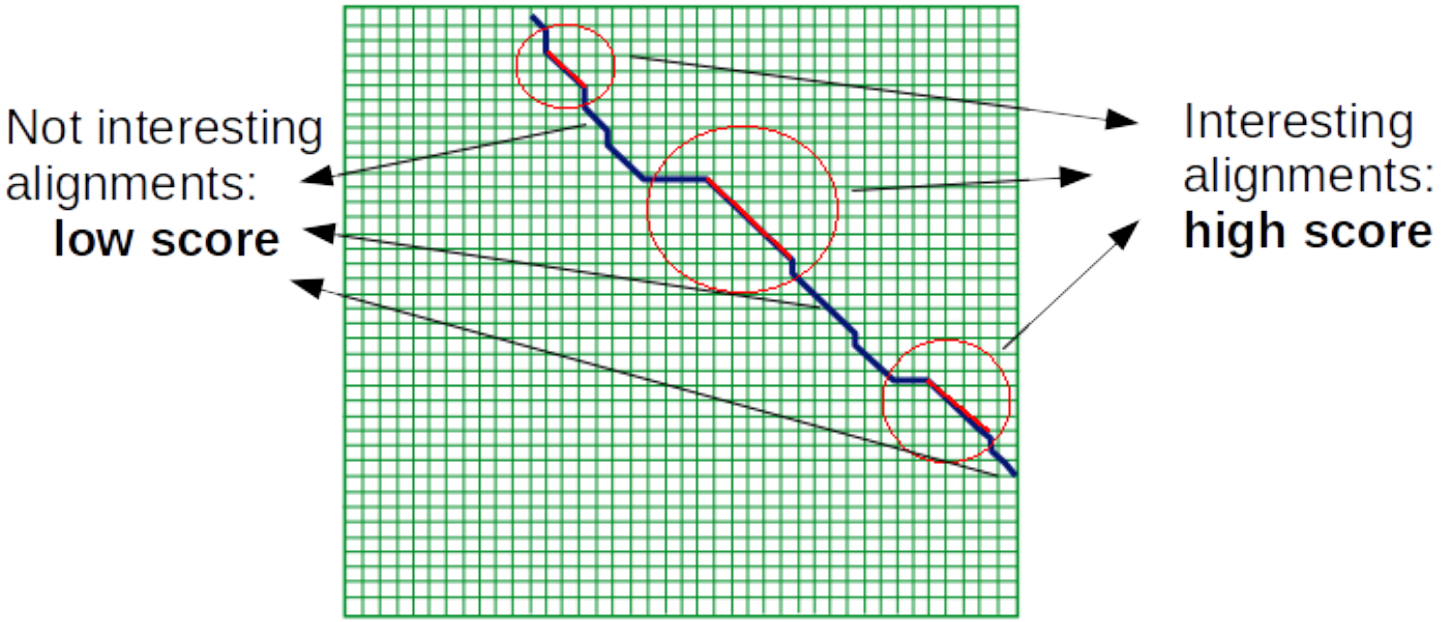
		A	C	A	C	T	G	A	T	C	G
	0	0	0	0	0	0	0	0	0	0	0
A	0	1	0	1	0	0	0	1	0	0	0
C	0	0	2	1	2	1	0	0	1	1	0
A	0	1	1	3	2	2	1	1	0	1	1
C	0	0	2	2	4	3	2	1	1	1	1
T	0	0	1	2	3	5	4	3	2	1	1
G	0	0	0	1	2	3	6	6	6	6	6

- Optimal alignment:

```
ACACTGATCG
ACACTG----
```

The Smith-Waterman Algorithm

- Sometimes even semiglobal alignments do not offer the flexibility needed in a biological sequence search.
- For example: find regions of interest for an unknown organisms vs a known genome



- Semiglobal alignment will not suffice, since each nonmatching position will be penalized
- The appropriate tool for this sort is a **local alignment**

Local Alignments

- Local alignments will find the best matching subsequences within the two search sequences.
- For example: AACCTATAGCT and GCGATATA (match=1, mismatch=-1, gap = -1)
  - A semiglobal alignment:

```
AAC-CTATAGCT
-GCGATATA---
```

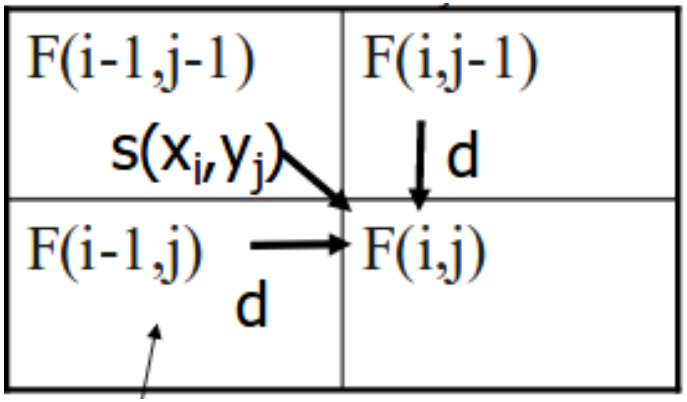
  - A poor alignment: many mismatches and gaps
  - But it shows and interesting matching region: TATA

How to modify the algorithm to identify only interesting matching regions?

Ignoring mismatches and gaps before and after the matching region

Smith and Waterman Local Alignments Algorithm

- Smith and Waterman introduced this algorithm in 1981, and is a fundamental technique in bioinformatics (base of the BLAST tool)
- To perform a local alignment we modify our global algorithm allowing a four option when filling in the partial scores table:



  - First rule: diagonals (match or mismatch)
  - Second rule: vertical (gap in first sequence)
  - Third rule: horizontal (gap in second sequence)

Fourth rule: Place a zero in any position in the table if all of the other methods result in scores lower than zero

Local Alignment Example:

Sequences: AACCTATAGCT and GCGATATA
Scores: math=+1, mismatch=-1, gap=-1

- Semiglobal aligment:

```
AAC-CTATAGCT
-GCGATATA---      Score = 5-7 = -2
```

- Local alignment:

		A	A	C	C	T	A	T	A	G	C	T
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	1	0	0
C	0	0	0	1	1	0	0	0	0	0	2	1
G	0	0	0	0	0	0	0	0	0	1	0	1
A	0	1	1	0	0	0	1	0	1	0	0	0
T	0	0	0	0	0	1	0	2	1	0	0	1
A	0	1	1	0	0	0	2	0	3	2	1	0
T	0	0	0	0	0	1	1	3	2	2	1	2
A	0	1	1	0	0	0	2	2	4	3	2	1

```
TATA
TATA              Score = 4
```

Assignments

- Implements the three algorithms:
  - Global
  - Semiglobal
  - Local
- Study the BLAST algorithm
- Study the FASTA algorithm

References

- Krane (2003). Fundamental Concepts of Bioinformatics
- Pevsner (2015)-Bioinformatics-and-Functional-Genomics-3ed.pdf